Research Directions for AI in Computer Games

Chris Fairclough, Michael Fagan, Brian Mac Namee, Pádraig Cunningham

Department of Computer Science,

Trinity College, Dublin 2, Ireland.

Abstract

The computer games industry is now bigger than the film industry. Until recently, technology in games was driven by a desire to achieve real-time, photo-realistic graphics. To a large extent, this has now been achieved. As game developers look for new and innovative technologies to drive games development, AI is coming to the fore. This paper will examine how sophisticated AI techniques, such as those being used in mainstream academic research, can be applied to computer games and introduce three projects doing just that.

1. Introduction

The computer games industry is now bigger than the film industry **[7]**. Until recently, technology in games was driven by a desire to achieve real time photo realistic graphics. To a large extent, this has now been achieved. At least, it will no longer be huge leaps in graphics technology that make one game stand out in the manner that Doom (www.idsoftware.com) stood out when it was first released in 1993.

This leaves the stage set for another aspect of gaming technology to move to the forefront. One of the real contenders for this role is AI. Although graphics technology allows the creation of games set in environments that look incredibly realistic, the behaviour of computer controlled characters, referred to as Non Player Characters (NPCs), often leads to a shallow and unfulfilling game experience. The application of sophisticated AI techniques to the control of NPCs could rectify this situation and create more immersive games.

This paper will begin with an overview of the more important genres of computer games available today, and some indication of the roles we see for AI in these games. From this we will explore the current state of the art of game AI. This will include a review of both the techniques being used in commercial games, and those being pursued by academic research projects. Section 4 will discuss

the merits of game AI as a research topic, and outline some of its unique challenges. Finally, we will discuss the projects being pursued as part of the TCD Game AI project and introduce the techniques which we feel will be used in the next generation of game AI.

2. The role of AI in different game genres

Before embarking on a discussion of the different game genres on the market today, a glaring contradiction needs to be resolved. A large, and ever growing, body of research work into computer implementations of classic games, such as chess, Go and Othello, already exists. When we refer to computer games, we are not referring to games such as these. Rather, we refer to what might be more familiarly termed video games - games made specifically to be played on computers. Further, little of the research into classic games is applicable to the games considered by this project. The main reason for this is that the number of degrees of freedom in modern video games is far beyond that of classic games.

What follows is a description of some of the more important genres of computer games on the market today, and pointers to some of the interesting roles for the application of AI to these games. This discussion will loosely follow a similar discussion given in **[7]**.

2.1 Action Games

Action games are the most popular game genre on the market today. The basis of the games can change from conquering an alien horde single handed with just your trusty side arm, to Mad-Max style, post-apocalyptic vehicle based carnage. The game-play, however, remains much the same – high adrenaline action where the aim of the game is to shoot everything that moves. Today's typical action game takes place in a fully rendered 3-d environment viewed from a first person perspective, and populated by countless varieties of cannon fodder upon which to unleash your fury through a wide range of exotic weaponry.

It is in creating more intelligent opponents that the most obvious possibilities for integration of sophisticated AI arise. Currently, the trend is to use schedule based finite state machines (FSMs) **[12]** to determine the behaviour of the player's adversaries. Although this has been achieved to very good effect (1999's Game of the Year, Half-Life (www.valve.com) astounded game players with squad based tactics, and enemies with incredibly realistic sensing models), FSMs are by their nature very rigid and, behave poorly when confronted by situations not dreamt of by the designer.

A number of games (Opposing Force (www.valve.com), the sequel to Half-Life, stands out as a notable example) have also made impressive use of partners and support characters that assist the player throughout the game. Building upon this idea, some recent games have cast the player as a

member of a squad or team **[5]**. Notable examples include Tom Clancy's Rainbow Six: Rogue Spear (www.redstorm.com/rogue_spear) and Star Trek Voyager: Elite Force (www.ravensoft.com/eliteforce). This is another area in which there is a real opportunity for the further application of sophisticated AI.

2.2 Adventure Games

Visually, the adventure game has changed dramatically since "Adventure" was created by Willie Crowther and Don Woods in the early seventies. The basis of the genre, however, has remained much the same. Gameplay involves the player moving around a restricted locale, solving puzzles and interacting with characters in an attempt to further a story line.

While the original examples of this genre were text based (commands were given through the player typing basic English commands – "eat the peach", "enter building", "open door" etc.), nowadays they are graphically stunning and input is given in a variety of novel ways – the most common being the use of the mouse to direct the player's character (from which came the name "point and click adventure"). Classic examples of this genre include the Monkey Island (www.lucasarts.com) and the Gabriel Knight (www.sierrastudios.com) series.

Two interesting applications of AI to the genre are the creation of more realistic and engaging NPCs and maintaining consistency in dynamic storylines.

2.3 Role Playing Games

Often seen as an extension of the adventure game style, role playing games (RPGs) stem from the popular Dungeons & Dragons **(**www.playdnd.com) paper based games that originated in the 1970's. Over the past two decades the computer versions of these games have metamorphosed from being mostly text based to the beautifully rendered, hugely involved games available today. Baldur's Gate (www.interplay.com/bgate) was a turning point for the genre. The level of detail in the Baldur's Gate world involves complexity far beyond anything seen before, with completion of the game involving over 100 hours of gameplay. RPGs see the player taking on the role of an adventurer in an exotic, mythical world, where gameplay consists of questing across the land, engaging in a mixture of puzzle solving and combat. Interactions with NPCs and an intricate plot are also important in the genre.

The differences between RPGs and adventure games arise from the scope involved. RPGs take place in far larger worlds and the player has more freedom to explore the environment at their own pace. Also, underlying RPGs is some rule set stemming from the original, and quite complex, Dungeons & Dragons rules.

The RPG format offers the same kind of challenges to the AI developer as the adventure game. However, extra complication is introduced due to the amount of freedom afforded to the player. Maintaining story consistency becomes a bigger issue and the level of sophistication required in an RPG's NPCs is beyond that required in adventure games.

2.4 Strategy Games

Strategy games cast a player in charge of a range of military units, controlled from a "gods-eye-view", which must be sent into battle against one, or more, opponents. Typically resources (such as gold, wood and stone) must be harvested in order to create units or construct buildings. This management of the construction of units is at the core of strategy gameplay, as different units perform to varying levels against each other, and come at varying costs. More recently, diplomacy has also featured strongly in strategy gameplay. Strategy games on the market today are an even mix between mythical, fantasy and science fiction campaigns; and recreations of historical battles.

Two distinct classes of game have emerged from the strategy genre. Turn based strategy (TBS) games involve each player taking their turn to move units, order production, mount attacks and so on, one after another. The Civilization (www.firaxis.com/civ3) series is the definitive example of this kind of game. Real time strategy (RTS) games, as the title suggests, take place in real-time with players moving units, ordering production etc. in parallel. The Age of Empires (www.ensemblestudios.com) and Command & Conquer (www.westwood.com) series, along with Total Annihilation (www.cavedog.com), stand out as fine examples of this genre.

One other sub-genre to spawned by the strategy game, is that of the God game. These cast the player in the role of a protective deity. The main factor distinguishing God games from strategy games is in the manner in which the player can take action in the environment. The player has the ability to manipulate the environment – for example to raise or flatten mountains to make the land more hospitable, or to unleash the fury of a hurricane or earthquake – and units are controlled less directly than in strategy games. Classic examples of this genre include SimCity (www.simcity.com), the Populous series (www.populous.net) and, the recently released Black and White (www.lionhead.co.uk).

AI in strategy games needs to be applied both at the level of strategic opponents and at the level of individual units. AI at the strategic level involves the creation of computer opponents capable of mounting ordered, cohesive, well planned and innovative campaigns against the human player. This is very challenging as players quickly identify any rigid strategies and learn to exploit them. At the unit level AI is required in order to allow a player's units to carry out the player's orders as

accurately as possible. Challenges at unit level include accurate path finding and allowing units a degree of autonomy in order to be able to behave sensibly without the player's direct control.

2.5 Others

Of course, just like any attempt at categorization, not all computer games fit neatly into one of the niches defined above. There is a large amount of overlap between the different categories – the hugely successful Diablo II (www.blizzard.com), for example, is considered an RPG, but a huge amount of the gameplay is made up of action combat sequences so, could it not also be considered an action game? Similarly, from time to time, a completely original title is released that simply defies categorization. One such example is the Sims (www.thesims.com) which was one of the shock successes of 1999. This game has been described as many things including an interactive soap opera and a dollhouse simulator. In spite of this, forming a categorization such as that above is useful in an attempt to clarify some of the basic requirements of game AI.

3. The Current State of the Art

In this section, we will describe the AI techniques that are currently being used in commercial games and some of the more interesting research efforts being undertaken in the domain.

3.1 The State of the Art within the game industry

AI, as used by commercial games developers is simplistic in comparison to the techniques being used in mainstream academic research and industrial applications[1]. Some of the more important reasons for this lack of sophistication include:

- a lack of CPU resources available to AI in games (up to the year 2000, typically about 10% of processor cycles **[18])**
- a suspicion in the game development community of the effects of using non-deterministic methods, e.g. neural networks
- a lack of development time – AI is usually only added to a game after most of the other sections of the game (e.g. the graphics engine) are complete
- a lack of understanding of advanced AI techniques in the game industry

---

[1] That is not to say that AI in some games has not been very impressive. The techniques being used are simplistic in comparison to those being used by academic researchers in such fields as machine learning and robotics.

- the fact that efforts to improve the graphics in games overshadowed all else, which led to a lack of research into other areas, particularly AI

This has led to the emergence of a number of very well established, well understood and robust techniques that are in wide use by game developers **[18]**. These include FSMs and their close relation Fuzzy State Machines **[12]**, the A* path finding algorithm **[16]**, and a number of A-Life techniques including Craig Reynolds' flocking algorithms **[11]**.

The fact that a number of core techniques are being repeatedly used in game design has led to a number of attempts to integrate them into a generic systems development kit (SDK). For example, Mathématiques Appliquées' *DirectIA* (www.animaths.com) and Louder Than a Bomb's *Spark!* (www.louderthanabomb.com). SDKs, however, have failed to take hold in the industry and Woodcock **[18]** suggests current SDKs lack of flexibility as the main reason for this.

Another well used technique in game AI is simply to cheat. This is particularly easy in some game genres. For example, in action games computer opponents can have perfect aim or the ability to see through walls and so, track a player. Similarly, in strategy games the computer opponent might be able to produce exactly the units needed without having to engage in the complicated resource management faced by a player.

Cheating as a technique is very processor efficient and can be very successful. However, it has one major drawback. If cheating is done badly and is noticed by the player it ruins the illusion of playing against an equally matched opponent, and destroys any sense of immersion built up by the game. This leads to a very unfulfilling game experience and so should be avoided.

To conclude, commercial game AI is dominated by a small number of simple, deterministic and processor efficient techniques that are very well understood and repeatedly used by the game development community.

3.2 Academic Research in Game AI

Academic research into AI for games has been rare over the past number of years, however the level of interest is growing. A number of research efforts are currently underway and courses are being offered in some US universities. Much of the research being undertaken has emerged from work conducted with military institutions. Many of the goals are similar and so there is a large crossover of techniques. One such effort is the *Soarbot* project **[17]** in which agents have been created to play the 3D action game Quake (www.idsoftware.com) using the rule based SOAR architecture. Forbus et al. **[4]** describe another interesting research project in which a military

system designed to analyse terrain in order to plan attacks **[3]** is being adapted for use in strategy games.

Another area in which a lot of work is being transferred to computer games is in the area of computer based story telling, a well-established research area. The Oz Project **[1]** is a successful research initiative that has been applying agent based AI techniques to the task of maintaining interactive stories. The Excalibur Project **[10]** is another effort that is concerned with creating agents to populate virtual game worlds.

Finally work undertaken as part of the RoboCup robot soccer tournament (www.robocup.org) offers a number of insights into problems similar to those arising in computer games.

4. Game AI as a research topic

Some might say that research into AI for computer games is not the most noble undertaking. We would argue, however, that computer games offer an accessible platform upon which serious cognitive research can be engaged. Laird and van Lent **[7]** go so far as to suggest that computer games are the perfect platform upon which to pursue research into human level AI. What follows are a number of the reasons we believe that computer games are a useful, and potentially rewarding research area.

In section 3 we put forward a number of reasons as to why AI in computer games was at present simplistic. The majority of these are no longer valid. One of the main factors holding back game AI development was the lack of available CPU resources. With more and more graphics processing moving to specialised graphics hardware, this is no longer an issue. Similarly, we mentioned that the games industry's focus on graphics technology has held back AI development. Graphics have now reached such a level that visually stunning games are the norm rather than the exception, causing games developers to look for innovations, other than superior graphics, to become the selling points of their games. This will inevitably lead to more time for research into areas such as AI. Thus, the stage is set for the emergence of superior AI to become the key feature in future game releases.

In terms of a research problem, AI for games is unique in the challenges it offers. One only has to consider the discussion of the many roles for AI in games given in section 2 to realise the wide range of problems arising. Games take place in dynamic complex worlds in which complex decisions, often based on partial knowledge must be made. This reads as a shopping list for the conditions required to formulate really hard AI problems.

Laird and van Lent **[7]** suggest a number of reasons why game AI is an attractive research area. These include the fact that the increasing realism in computer games and the fact that many game manufacturers are creating games with hooks to allow people to modify the game (known as mods), makes them an attractive alternative to expensive home grown simulations. Also, computer game worlds are reaching a level of complexity comparable with the real world, allowing simulations which concentrate on cognitive issues, without the extra burden of using unreliable physical sensors, and motor systems such as those used in robotics.

Finally, as was previously mentioned, computer games are now a multi-million dollar worldwide industry. This means that a direct route exists from research projects into viable commercial enterprises. Coupled with this is the fact that commercial games companies run to frantic schedules. Because of this, they are reluctant to spend time investigating riskier techniques for fear that they might not work out, and so precious development time will be lost. As researchers, we have the luxury of being able to research a technique without the burden of pressure that comes with having to produce a commercially successful product. If research shows a technique is not suitable to a particular task, that is still a valuable a piece of research, and time well spent. In the fast paced, bottom line driven world of commercial games development, this is certainly not the case.

The main drawbacks to research work on AI in games are the lack of formal structure surrounding the subject, and the amount of suspicion, and degree of secrecy amongst games companies. No formal journals exist and, apart from a few small conferences (for example the annual AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment), there is little contact between researchers and games developers. This results in researchers having to rely on a small number of well run web sites (Steve Woodcock's www.gameai.com being one of the better ones) in order to discover exactly what techniques are currently being used.

5. The TCD Game AI Project

The TCD Game AI project is concerned with applying sophisticated AI techniques to the domain of computer games. The main focus is on the creation of more interesting NPCs, and more immersive game experiences. We will conclude this paper with a short description of three projects that have been selected to achieve these goals.

5.1 Case-Based Plan Recognition Used by Non-Player Characters for Intelligent Interaction with Human Players

In the modern gaming arena, the average gamer is becoming increasingly adept at predicting which NPC scripts are to be executed next. The extensive use of FSMs is the main culprit in providing this

ease of prediction. This is because FSM's are deterministic, with rules governing what NPCs do in specific situations. In simple terms, the behaviour of the NPC is governed by a set of rules, each rule having a condition and some action(s) to execute when this condition is satisfied by the state of the world. Therefore, as a player interacts with an NPC he gets to know the behaviour of the NPC and can predict what it will do next and can develop a plan to despatch the NPC. Wouldn't it be interesting if the tables could be turned so that the NPC can predict what the player is going to do next?

Case-Based Plan Recognition **[6]** can be used in this realm as a prediction mechanism. As a player plays the game, a case-base of the player's plans and goals is built up. Then, while the player is playing, the NPC tries to predict the player's next move by *recognising* the plan they are currently executing. This is done by finding the best match in the case-base for the current partially executed plan. Then this retrieved case is used to predict what the player will do next.

There are two research issues here. First, there is the problem of identifying the plan that the player is currently executing. Then there is the question of how best to respond to this information. As a first step in this research a Space Invaders applet has been developed that will act as a test bed for plan recognition (http://www.cs.tcd.ie/Michael.Fagan/). With this applet we can capture the players actions as a click stream and analyse this to determine what plan the player is currently executing. A case-base of plans is built up and this is used to recognise the plan the player is currently executing. Clearly it is not in the spirit of Space Invaders for the actions of the invaders to change based on what they predict the player will do next so we will need to move to another game environment to research the question of how to respond to the predictions of the players behaviour.

5.2 Proactive Persistent Non Player Characters

"don't let monsters outside of the player's PVS [Potentially Visible Set] *act up, or most of the interesting things will happen before the player gets there"*

Code Comment from Half-Life SDK[2]

The above quote comes from a code comment in the Half-Life Systems Development Kit and represents an attitude behind a lot of current game design. The essence of this is that games are designed only to do interesting things in the areas of the game environment in which the player is

[2] The Half Life Systems Development Kit can be found at www.planethalflife.com/half-life/files/

currently active[3]. This approach is quite acceptable for many games. In particular for action games, which are heavily scripted and event driven, it would be pointless to have much going on before the player arrives at a location. However there are some games, particularly adventure games and Role Playing Games (RPGs), in which a virtual world, populated by agents that act autonomously - irrespective of the player's location - would greatly add to the sense of immersion.

One might well ask, do today's games not already offer this kind of environment? RPG and adventure games on the market today do have a vast range of NPCs with which the player can interact. However, these NPCs are not modelled until the player arrives at their location and then, either await the player's involving them in some interaction, or play through a scripted sequence. This kind of character can be denoted a *static agent*.

What this work proposes is a virtual world populated by *Proactive Persistent Agents (PPAs)* **[8]**. These are NPCs with needs, beliefs and desires of their own. From these, would arise goals which the characters would pursue, of their own accord and in real time. The persistence of these agents refers to the fact that they are always modelled, at least to some extent, regardless of the player's location in the game world.

Another important aspect of NPC design that is often overlooked and which will be included in the PPA scheme, is the characters' ability to form and engage in relationships with players and other NPCs. It is suggested in **[15]** that, as they are made today, computer games are quite limited in their appeal. For this to change a shift will have to occur towards what are referred to as "character centric" games. Such games place a much stronger emphasis on character interactions as the main focus of game play. For this to be achieved successfully NPCs' social abilities must go beyond what has been achieved to date. This requirement is very well illustrated by the following quote from **[2]**:

"The fact is, in most computer role-playing games, the NPCs don't give any semblance of intelligence," he continues. *"Say you're in a bar and you throw your beer at the bartender one day. The next day you go back, and he's just as happy to see you. That shouldn't happen."*

5.3 An Interactive Story Engine

The field of interactive stories has its roots in the arts of oral storytelling and theatre, and ideas from these fields have been incorporated into new forms of storytelling using the computer as a medium. Amongst the earliest projects based on the use of a computer storyteller was 'Tale Spin' by James

[3] This is an interesting version of the old philosophical question "Does a tree falling in the forest make a sound if there is no-one there to hear it?" In reality most people would believe it would, whereas in current computer games it most definitely would not!

Meehan[4] **[9]**, which composes stories based on giving each character some goals to achieve and forming plans for each of the characters. A very simple example output:

One day Wilma was very thirsty. Wilma wanted to get near some water. Wilma flew from her nest across the meadow through a valley to the river. Wilma drank the water. Wilma wasn't thirsty anymore. . . etc . . .

Since an early interest in the topic was quashed in the 70's, research in computer storytelling was non existent until relatively recently. Carnegie Mellon University in Pittsburgh was a hotspot of interest in the field in the early 90's, spearheaded by Joseph Bates **[13]**, **[14]**. The OZ project[5] at CMU consists of a group of researchers interested in diverse aspects of the field, and has spawned some companies that are developing some very interesting products, for example www.ottoandiris.com.

This part of the TCD project will consist mainly of developing a 3D game engine which is populated by believable characters, and which has a story director agent (SD) that has some control over these characters. This agent pushes the characters into following a dynamic, coherent plotline which changes according to the player(s)' actions in the game world.

The SD is an agent that monitors the state of the story world and each character in it, and makes decisions about how the story should develop. These decisions are made according to a case based planning system that tries to recognise what story structure the player's actions would fit into. The SD affects the story world by assigning actions to characters.

6. References

[1] Bates, J., The Nature of Character in Interactive Worlds and The Oz Project, Technical Report CMU-CS-92-200, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. October 1992.

[2] Coco, D., Creating Intelligent Creatures, *Computer Graphics World,* July 1997.

[3] Forbus, K., Nielsen, P. and Faltings, B., Qualitve Spatial Reasoning: The CLOCK Project, Artificial Intelligence, 51 (1 – 3), October 1991.

---

[4] An evaluation of Tale Spin can be found at www.media.uwe.ac.uk/masoud//author/story/story-2.htm

[5]Information about the Oz project can be found on Carnegie Melons web site at www.cs.cmu.edu/afs/cs.cmu.edu/project/oz/web/

[4] Forbus, K. D., Mahoney, J. V. and Dill, K., How Qualitive Spatial Reasoning Can Improve Strategy Game AIs, Proceedings of the 2001 AAAI Spring Symposium on Artificial Intelligence and Computer Games, 2000.

[5] Gibson, C. and O'Brien, J., The Basics of Team AI, Proceedings of the 2001 Game Developers Conference, pp. 323 – 332, 2001.

[6] Kerkez, B., Cox, M. T., Incremental Case-Based Plan Recognition Using State Indices, to be presented at The International Conference on Case-Based Reasoning, Vancouver, Canada, August, 2001.

[7] Laird, J. E., van Lent, M., Human-level AI's Killer Application: Interactive Computer Games, Proc. AAAI 2000, pp. 1171-1178. AAAI Press / The MIT Press, 2000.

[8] Mac Namee, B., Cunningham, P., A Proposal for an Agent Architecture for Proactive Persistent Non Player Characters, Departmental Technical Report TCD-CS-2001-20, Trinity College Dublin, 2001.

[9] Meehan, J., The metanovel: writing stories on computer. University Microfilms International, 1977.

[10] Nareyek, A., Intelligent Agents for Computer Games, *In Proceedings of the Second International Conference on Computers and Games (CG 2000)*, 2000.

[11] Reynolds, C. W., Flocks, Herds and Schools: A Distributed Behavioural Model, Computer Graphics, 21(4), July 1987, pp. 25 –34.

[12] Russell, S. and Norvig, P., Artificial Intelligence: A Modern Approach, Prentice Hall, 1995.

[13] Scott Reilly, W. and Bates, J., Building Emotional Agents, CMU-CS-92-143, 1992

[14] Smith, S. and Bates, J., Towards a Theory of Narrative for Interactive Fiction, CMU-CS-89-121, 1989.

[15] Stern, A., AI Beyond Computer Games, *Presented at AAAI AI and Interactive Entertainment Spring Symposium 2000*.

[16] Stout, B., Smart Moves: Intelligent Path-Finding, Game Developer Magazine, October 1996.

[17] Van Lent, M., Laird, J. E., Buckman, J., Hartford, J., Houchard, S., Steinkraus, K. and Tedrake, R., Intelligent Agents in Computer Games, Proceedings of the National Conference on Artificial Intelligence, July 1999, Orlando, FL, pp. 929-930.

[18] Woodcock, S., Game AI the State of the Industry, Game Developer Magazine, August 2000.